
xCall[®]

Technische Dokumentation zu Version 4.x

Autor: Martin Roth
Erstelldatum: 14.08.2008
Version: 1.4





Zweck 3

Schnittstellenarchitektur 3

 Outbound-Schnittstellen..... 3

 Outlook 3

 TwixTel 3

 Excel 3

 CSV 4

 ODBC 5

 Starter..... 5

 Inbound Schnittstellen..... 6

 Aufruf mit Parameter 6

 TapiRequestMakeCall 6

Interaktionen der Outbound Schnittstellen 7

Zweck

Zweck dieses Dokumentes ist die Beschreibung der Schnittstellen von xCall, um es zusammen mit anderen Anwendungen verwenden zu können.

Das Dokument richtet sich an System-Entwickler und setzt gute Kenntnisse von Windows voraus. Je nach Schnittstelle werden auch weitere (Programmier-) Kenntnisse vorausgesetzt.

Schnittstellenarchitektur

Grundsätzlich existieren zwei Arten von Schnittstellen: Die in xCall eingebauten Schnittstellen zu anderen Applikationen (Outbound) und die Schnittstellen von anderen Applikationen zu xCall (Inbound).

Outbound-Schnittstellen

Outbound-Schnittstellen werden typischerweise benutzt, um bei einem Anruf die Telefonnummer zu einem Namen aufzulösen. Je nach Schnittstelle können dabei auch andere Kontakt-Daten (z.B. Adresse) ins Spiel kommen.

Outlook

Die Outlook-Schnittstelle ist zweifellos die mächtigste aller in xCall eingebauten Schnittstellen.

Die Schnittstelle verwendet den installierten Outlook-Client um an die Kontakt-Daten zu kommen. Im Hintergrund wird Outlook unter dem momentan eingeloggten User gestartet (falls nicht schon passiert), so werden alle Einstellungen für diesen User aktiv. So wird auch ein eventuelles Exchange-Profil übernommen, inklusive öffentlicher Ordner, etc.

Weitere Details zu dieser Schnittstelle finden Sie im xCall Manual.

TwixTel

Diese Schnittstelle wird durch ein von der Firma Twix geliefertes DLL realisiert. Das DLL braucht lediglich den Pfad zur Daten-CD als Laufwerksbuchstabe oder UNC.

Excel

Die Excel Schnittstelle ist eigentlich drei Schnittstellen:

Telefonbuch:

Beim Starten von xCall werden alle Einträge der spezifizierten .xls Datei eingelesen. Oder genauer gesagt: xCall liest eine Zeile nach der anderen bis mehr als 50 Zeilen keine Telefonnummer enthalten. Es dürfen also leere Zeilen vorhanden sein. Die Tabelle muss dabei ein definiertes Format einhalten, ein Beispiel dafür finden Sie unter www.xCall.ch.

Journal:

Nach Beendigung eines Anrufes wird der spezifizierten Excel-Datei vor der 3. Zeile eine neue Zeile eingefügt, deren Felder mit den Anruf-Details gefüllt werden. Die Details sind die gleichen wie im Journal-Fenster. Die Excel-Datei kann auch von mehreren Usern gleichzeitig benutzt werden (Gruppenlaufwerk), wobei jedoch nur ein User gleichzeitig darin schreiben darf.

Kostenstelle:

Erlaubt es, die in der Excel-Tabelle angegebenen Kostenstellen per Bezeichnung zu wählen.

CSV

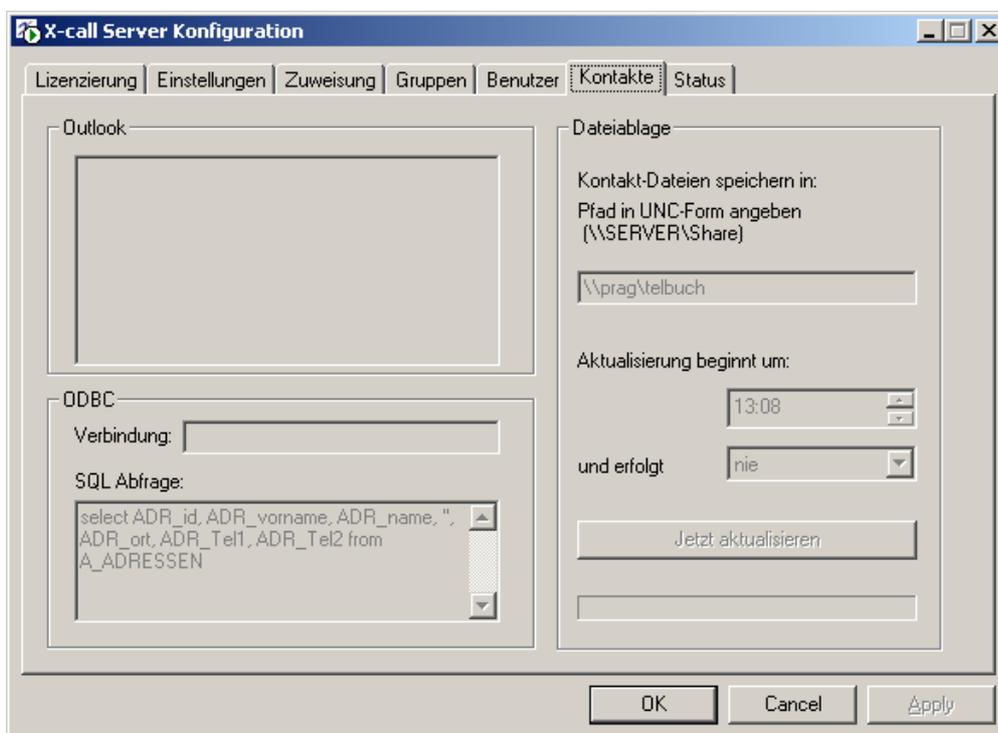
Der xCall Client kann Kontaktdaten aus einer CSV-Datei lesen. Die CSV-Datei muss in einem festgelegten Pfad liegen und ODBC.csv heissen. Die erste Zeile enthält die Felddefinitionen und wird in der aktuellen Version noch ignoriert.

Die Felder müssen in einer spezifischen Reihenfolge auftreten:

1. Datenbank-ID (Primarykey, welcher von der Starter-Anbindung verwendet werden kann)
2. Vorname
3. Nachname
4. Firma
5. Telefon 1 (Hauptnummer)
6. Telefon 2 (Geschäft)
7. Telefon 3 (Mobile)
8. Telefon 4 (Zuhause 2)
9. Telefon 5 (Geschäft 2)

Per Default müssen die Felder mit einem Komma getrennt sein. Soll ein Semikolon verwendet werden, so muss in der ersten Zeile der CSV-Datei ein Semikolon vorkommen.

Die Datei wird vom Client im Verzeichnis gesucht, deren UNC-Pfad im Server eingestellt wird. Siehe „Dateiablage“ im Screenshot:



ODBC

Die ODBC-Schnittstelle, die auf dem xCall Server läuft, lässt xCall Kontaktdaten aus ODBC-fähigen Datenbanken lesen.

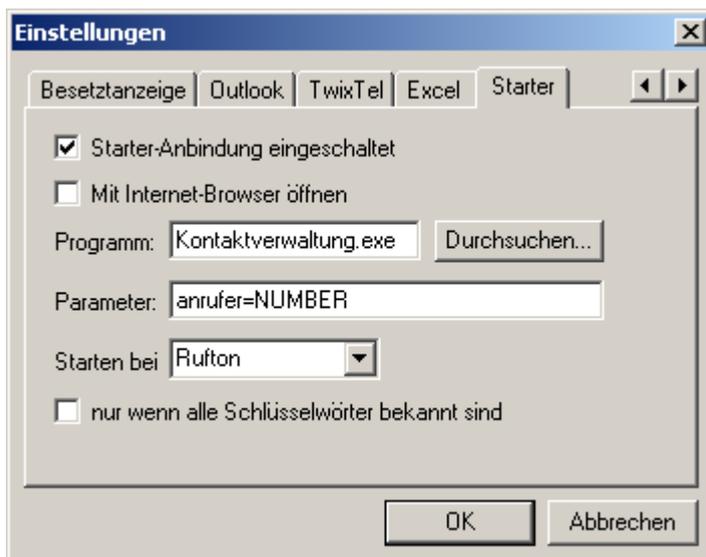
Es muss der Connection-String und die SQL-Query spezifiziert werden. Der Connection-String muss im ODBC-Format sein, z.B. „DSN=MeineDatenbank;“, falls eine DSN definiert wurde. Andere String finden Sie auf <http://www.connectionstrings.com/>

Die SQL-query muss die Felder in der gleichen Reihenfolge liefern wie unter CSV beschrieben.

Starter

Die Starter-Anbindung wird verwendet, um eine beliebige ausführbare Datei zu starten. Dies ist typischerweise ein .exe, kann aber z.B. auch ein .bat oder sein. Dabei können der so gestarteten Anwendung beliebige Parameter mitgegeben werden. xCall erlaubt es nun auch, Parameter dynamisch zu setzen, so kann der zu startenden Applikation z.B. die Telefonnummer des Anrufers übergeben werden. Dabei werden definierte Schlüsselwörter durch die aktuellen Werte ersetzt.

Beispiel:



Bei einem Anruf wird das Programm Kontaktverwaltung.exe mit dem Parameter anrufer=x aufgerufen, wobei x durch die Telefonnummer des Anrufers ersetzt wird.

Liste der Schlüsselwörter:

NUMBER	Telefonnummer des Anrufers
CANONICAL	Nummer im kanonischen Format
ADDRESS	Interne Nummer des kontrollierten Endgerätes
STREET	Strasse
STREETNUMBER	Hausnummer
ZIP	Postleitzahl
CITY	Ort
FULLNAME	Vollständiger Name, evtl. mit Firma
FIRSTNAME	Vorname
LASTNAME	Nachname
COMPANY	Firma
REDIR	Umleitungs-Informationen der PBX. Zeigt bei ascotels auch die ARV-Informationen
IDENTITY	Datenbank-ID, falls eine CSV-Datei verwendet wurde

Falls "Mit Internet Browser" öffnen gewählt wird, wird nicht direkt ein Programm gestartet, sondern die Shell versucht, das unter Parameter eingegebene Dokument zu öffnen. Es hängt somit von der Shell ab, welche Applikation benutzt wird, um das Dokument zu öffnen. Im Grunde passiert das selbe, als wenn man im Windows Explorer auf das Dokument doppelklickt. Die Checkbox heisst "Mit Internet Browser" weil die Option meistens benutzt wird, um Intranet-Dokumente zu öffnen.

So können z.B. ASP-Seiten auf einem Intranet-Server programmiert werden, die per Query-String eine Telefonnummer in einer ODBC-Datenbank suchen.

Inbound Schnittstellen

Die Inbound Schnittstellen dienen dazu um xCall eine Nummer wählen zu lassen. Damit kann aus beliebigen Applikationen telefoniert werden.

Aufruf mit Parameter

Wird xCall.exe mit einem Parameter aufgerufen, so nimmt xCall an, dass es sich dabei um eine zu wählende Nummer handelt und wählt diese. Dabei wird sichergestellt, dass nur eine Instanz von xCall läuft, d.h. es wird kein zweites Fenster geöffnet.

Das einzige Problem dieser Schnittstelle ist, dass nicht bekannt ist, in welchem Verzeichnis xCall installiert ist. Per Default installiert die Setup-Routine xCall in %ProgramFiles%\xCall, aber dies kann vom User verändert werden.

TapiRequestMakeCall

Die Windows-Systemkomponente TAPI bietet den Befehl tapiRequestMakeCall an, welcher dazu dient, einen Anruf mittels einer registrierten Applikation durchzuführen. Beim Hochfahren registriert sich xCall und kann so tapiRequestMakeCall-Aufforderungen ausführen.

Um diesen Befehl zu nutzen, muss die aufrufende Applikation lediglich die Funktion `tapiRequestMakeCall` aus der `tapi32.dll` aufrufen und als ersten Parameter einen String mit der zu wählenden Nummer übergeben. Die restlichen 3 Parameter werden auf Null gesetzt.

Diese Schnittstelle läuft auf Terminal Server Umgebungen nicht.

Interaktionen der Outbound Schnittstellen

Falls mehrere Outbound Schnittstellen lizenziert wird, so werden diese mit einer bestimmten Logik miteinander verknüpft. Der Trigger für jede Schnittstelle ist wenn eine Telefonnummer des Anrufers durch den Telefonie-Treiber gemeldet wird, bzw. sich diese ändert (z.B. aufgrund eines Vermittlungs-Vorganges).

